

Remarks

This paper is responsive to the Office action mailed July 17, 2003. A request for the necessary extension of time in which to respond is filed concurrently herewith.

Figure 3 in the drawings was objected to for failing to include a reference numeral "285" appearing in the specification. The drawing has not been changed because it is correct and the specification is in error. Figure 3 correctly identifies a profiling agent by reference numeral "280" while the specification incorrectly identifies the same agent by reference numeral "285". The specification has been amended by rewriting the relevant paragraph to correct the erroneous reference numeral.

Copies of the subject application available to the attorney of record did not include a page 3 known to have been included in the application as prepared. Since that raised the possibility that the application submitted to the Office also failed to include a page 3, this amendment asks that the originally prepared page 3 be inserted into the application if it is not already there.

The inclusion of the page is considered to be appropriate as the page does not include any matter specific to the invention described and claimed in the application. Page 3 describes the use of known performance profiling technology for profiling a Java Virtual Machine (JVM) using a JVMPPI or Java Virtual Machine Profiler Interface. The material on page 3, while useful in gaining an appreciation of the present invention, is strictly prior art well known to those of ordinary skill in the art. No truly new matter is being added to the application.

All claims originally submitted in the subject application were rejected either under 35 USC 102 or 35 USC 103 based on supposed Admitted Prior Art described between page 10, line 21 and page 12, line 7 of the application.

When the Office action was sent to the originating attorney and the inventor for comment, they realized what had been characterized as "Prior Art" was in fact a description of nonpublic, preliminary work done by the inventor himself in the course of developing the invention covered in the subject application. An Inventor's Declaration is appended to this paper setting forth the

PATENT
IBM Docket No. GB9-2000-0047US1

circumstances and outlining the reasons why the material could not properly be characterized as prior art under the United States patent laws.

Several paragraphs in the specification have been rewritten to make it clear that they were intended to refer to the inventor's own work, which is not prior art for the subject invention. Figure 3 was intended to illustrate the inventor's own work, not the prior art. Figure 3 is amended to remove the inappropriate "Prior Art" legend. A clean copy of the amended Figure is attached to this amendment.

Since the prior art on which the only rejections of the claims were founded does not, in fact, exist, it is submitted that the rejections cannot appropriately be maintained and should be withdrawn. As all other objections have been remedied, it is believed this application is now in condition for allowance.

Respectfully Submitted,

Gerald R Woods

Gerald R. Woods, Reg. No. 24,144
Attorney of Record

IBM Corporation
T81/503
PO Box 12195
Research Triangle Park, NC 27709
919-(919) 543 - 7204
FAX 919-254-4330

advantageous features, including its portability, efficiency and security. The JVM emulates a system's hardware, by stepping through a program and carrying out each instruction contained therein. Thus, the JVM need
5 only be ported once for each operating environment and then all applications running on top of the virtual machine may execute unchanged. Furthermore, the JVM creates a layer of abstraction between a computer and the software being run on it, hence providing greater protection against malicious
10 viruses etc.

With Java's growing popularity, performance profiling tools, which in particular monitor the performance of the JVM and applications running on top of it, have become an
15 important issue. Sun Microsystems Java 2 software development kits (SDKs) support a performance profiling interface known as the Java Virtual Machine Profiling Interface (JVMPPI). This is an interface provided to external agents written in the 'C' programming language
20 which allows the agents to register with the JVM and receive notifications of events of interest to performance tools via this interface.

The JVMPPI represents a considerable advance over
25 previous Java profiling solutions as it offers a standard, extendible interface, agreed through consultation with Java tools vendors and other companies such as IBM. It defines a